

**Overview:**

In the Pet Car Alarm project, students build and program a simple feedback and control loop. Feedback and control loops are central to many industrial systems and consumer products. The system uses three input modules; Two temperature sensors and a Hall effect magnetic field sensor. The system has four outputs that include two white LED lights (headlights), 1 continuous servo motor (controls the window) and also makes use of the TI-Innovator Hub's built-in speaker (horn honking). The continuous servo requires more power than the calculator can provide. Therefore, an external USB battery is needed. A program is written to read all three input parameters and logically compare them with critical set-point values to determine when to turn on the alarm. The project is presented in a series of small challenges that build the knowledge and skills required for the final open-ended challenge. The final challenge also relies on the students' understanding of important biology and Earth science topics that are relevant to optimizing the system the students ultimately design and refine.

Note: This project assumes the students have a working knowledge of the programming concepts in the Digital Mood Ring project. Please refer to that project as a review of concepts. The materials for the Digital Mood Ring project can be found here - <http://bit.ly/2oqU1X8>

**Possible NGSS topics to explore with students:**

**Disciplinary Core Ideas:**

- HS-LS1-3 – Mechanisms of homeostasis
- MS-ETS1-1 to 1.4 – Engineering design
- MS-PS3-3 – Design that maximizes thermal energy transfer

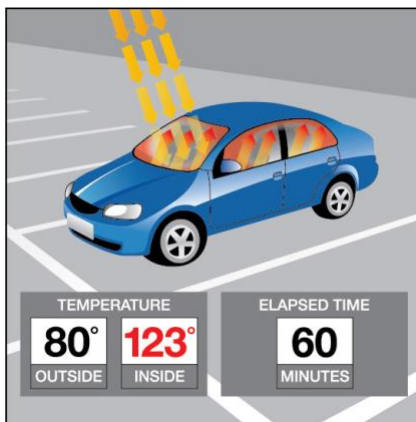
**Science & Engineering Practices:**

- Asking questions & defining problems
- Evaluate competing designs
- Constructing explanations & designing solutions

**Crosscutting Concepts:**

- Systems & System Models
- Cause & Effect
- Energy & Matter

**Background:**



Estimated Vehicle Interior Air Temperature v. Elapsed Time						
Elapsed Time	Outside Air Temperature (F)					
	70	75	80	85	90	95
0 minutes	70	75	80	85	90	95
10 minutes	89	94	99	104	109	114
20 minutes	99	104	109	114	119	124
30 minutes	104	109	114	119	124	129
40 minutes	108	113	118	123	128	133
50 minutes	111	116	121	126	131	136
60 minutes	113	118	123	128	133	138
> 1 hour	115	120	125	130	135	140

Inside Air Temperature (F)



**Light to Heat**

Each summer, there are horrifying stories of children and pets left in hot cars. Ultimately, most end up having a heat stroke and in many cases pass away. The inside of a car heats up so much faster than the outside due to the greenhouse effect. Rays of sunlight stream into the vehicle through the windows and strike, the light strikes the surfaces of the interior of the car. That visible light is absorbed and reradiated as infrared light. Infrared radiation has a larger wavelength than visible light. The infrared radiation is unable to escape back through the windows. The trapped radiation causes the temperature inside the car to rise faster than the outside temperature.

**Stayin' Cool!**

Mammals such as humans, dogs, and cats all have ways to regulate temperature, thermoregulation, to maintain homeostasis. Humans sweat to increase the removal of heat from the body through evaporation. Dogs usually pant to remove heat although they also have a small number of sweat glands in the pads of their paws. Cats will sprawl out on surfaces that are relatively cool to help remove body heat. They will lick their paws and rub the saliva on warmer parts of their bodies to increase evaporative cooling which is a similar mechanism as sweating in humans. When thermal regulation mechanisms are unable to maintain homeostasis, that mammal will go into heat distress.

**The problem with too much heat**

Heat distress leads to brain impairment, dehydration, heart failure, cell swelling, and protein denaturation... and potentially death. Conduct research into why mammals such as humans, dogs, and cats are unable to endure hot environments like closed cars in the summer for long periods. Then design a solution using technology to protect car occupants from heat distress. Refine your design until you have a working prototype. With your teacher's permission, compare your prototype with those of your classmates to determine which team has the "best" system.

<b>Command</b>	<b>Example</b>	<b>Behavior</b>
CONNECT <type> <number> TO <port>	Send "CONNECT TEMPERATURE 1 TO IN1"	Associates the first TEMPERATURE object with a temperature module plugged into port IN1 on the Hub.
SET <type><number> TO <value>.	Send "SET LED 1 TO ON"	Turns on LED 1. Other parameters may include BLINK <rate> TIME<duration> "SET LED 1 ON BLINK 3 TIME 20"
SET <type><number> TO <value> [<blink rate>] [<duration in seconds>]	Send "SET LED 1 ON BLINK 3 TIME 20"	Sets LED 1 to a blink rate of 3 times per second for 20 seconds. See Hub Settings menu for ON, OFF, BLINK, etc.
SET <type> <number> TO <value>	Send "SET SERVO.CONTINUOUS 1 CW 20 TIME 1"	Turns on the first continuous servo motor object at power 20 (range 0-100) in the clockwise direction (CW vs. CCW) for a time of 1 second.
Wait <number>	Wait 3	Pauses the program for a specified number of seconds. In this case 3 seconds.
READ <type> <number>	Send "READ TEMPERATURE 1"	Reads a measurement from the first temperature object.
Get <variable>	Get t	Stores the temperature measurement into a variable named t. The value stored will contain the measurement from the immediately preceding READ command. Note: a



		Get command must immediately follow a READ command.
DispAt <line #> , <"text"> , <variable name>	<code>DispAt 3, "Temperature = ", t</code>	When variable <i>t</i> has a value of 26, the following line is displayed on the calculator: Temperature = 26 (temp readings are in °C by default)
For <counter variable>,<start value>,<end value>,<step value>] <statements> EndFor	<code>For n,1,10   DispAt 3,n EndFor</code>	Runs For loop 10 times, starting at 1 and ending at 10. Executes the statement in the block each time, displays the value for the counter variable on row 3.
While <Boolean expression> <statements> EndWhile	<code>key:="" While key!="esc"   READ "TEMPERATURE 1"   Get t   DispAt 3,"Temperature= ",t   key:=getKey()   Wait 1 EndWhile</code>	The statements in the While loop are executed until the escape key is pressed. The While loop continues as long as the Boolean expression evaluates to "true". The variable <i>key</i> is set to an initial value of an empty string using the assign function, ":=". getKey() is a function that returns a string with the name of the last key pressed while a programming is running. In this program the value of getKey() is stored to the variable <i>key</i> .
<Boolean expression> and <Boolean expression>	<code>If t&gt;25 and m&lt;100 Then   Send "SET COLOR 0 255 0" EndIf</code>	When both expressions are true the "and" function is "true" and the statements are executed. Otherwise, the function returns false and the statements are skipped.
If <Boolean expression> Then <statements 1> Else <statements 2> End	<code>If m&lt;100 Then   DispAt 5,"Magnet is present" Else   DispAt 5,"Magnet is not present" EndIf</code>	The example decision tree has a Boolean expression with corresponding statements to execute if true. It also has an Else condition that executes corresponding statements when the Boolean expression is false. This Else condition ensures that a set of statements will always be executed. When this decision tree executes, focus proceeds from top-down. If the value of <i>m</i> is less than 100 the commands after Then are executed. In this case the output "Magnet is present" is displayed on row 5. If the value of <i>m</i> is greater than or equal to 100 the commands after Else are executed. In this case the output "Magnet is not present" is displayed on row 5.

See TI-Innovator Technology eGuide for more background [https://education.ti.com/html/webhelp/EG\\_Innovator/EN/index.html](https://education.ti.com/html/webhelp/EG_Innovator/EN/index.html)

**Setup Project:**

**Supplies:**

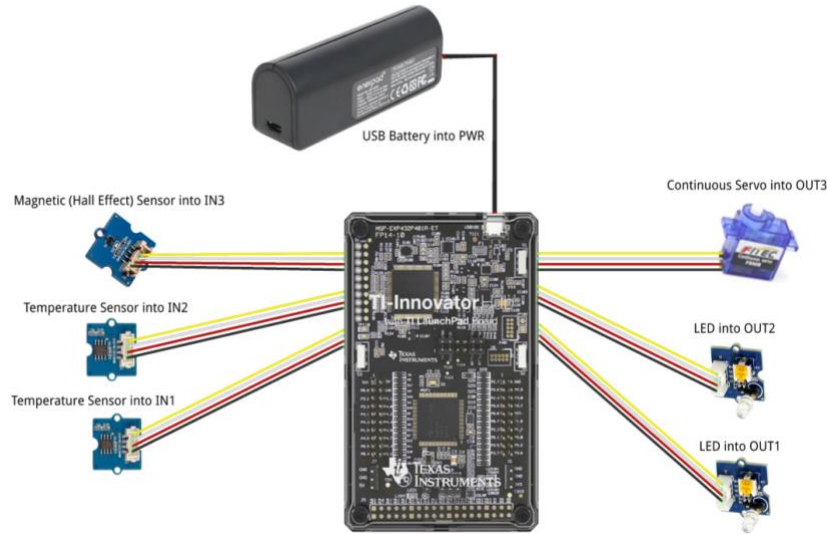


# Challenge: Pet Car Alarm Project

## TI-NSPIRE™ CX

# TI-INNOVATOR™ STEM PROJECT

## TEACHER NOTES



- Calculator
- Unit to Unit Cable
- TI-Innovator Hub
- Grove Temperature sensor x2 (inside and outside measurement)
- Grove Hall effect magnetic proximity sensor
- Grove White LED Light x2
- Grove Continuous Servo motor
- Grove Cable x5
- External USB Battery w/ Micro USB Cable
- Fashion Doll Car, shoebox, or another object to model a car
- Small toy pet
- Small magnet (ceramic, ferrite or ceramic-ferrite) for pet collar
- Small piece of clear plastic to model car window
- Cellophane tape to attach magnet to pet and window to motor
- Cling wrap (e.g. Saran) to enclose model car
- Safety scissors

### Student Activity:

### Teacher Notes:

**Challenge 1:** Write a program that will play two sounds for 1 second each in a For loop that repeats five times.

e.g. Send "SET SOUND 220"

### Teacher Guidance during Challenge 1:

- Use a For..EndFor loop to alternate between two different sound frequencies one octave apart. Loops are used to repeat a set of command. A For loop repeats a specified number of times. In TI-Basic the programmer defines a For loop with four inputs: a counter variable, a beginning value for the counter variable, an ending value for the counter variable and an optional step value variable.
- For...EndFor is found on the Program Control Menu
- Use the Wait command to determine how long each tone plays
- Example program:

```
Define pcl()=
Prgm
For n,1,10
  Send "SET SOUND 440"
  Wait 1
  Send "SET SOUND 880"
```

```
Wait 1
EndFor
EndPrgm
```

- Try adding a third tone in your program
- Try to find the limits of the built-in speaker by changing the frequencies

**Challenge 2a:** Use a For..EndFor to blink two external LED's.

**Teacher Guidance during Challenge 2:**

- The Hub Connect commands provide the hub with information about the type of device plugged into external ports and how to communicate with that device.
- Connect LED's to OUT 1 and OUT 2 on the Hub
- Use the Send "CONNECT LED # TO OUT #" command
- Use a For..EndFor command to make the LED's turn off and on
- Example program:

```
Define pc2a()=
Prgm
Send "CONNECT LED 1 TO OUT 1 "
Send "CONNECT LED 2 TO OUT 2 "
For n,1,30
  Send "SET LED 1 ON"
  Send "SET LED 2 ON"
  Wait 1
  Send "SET LED 1 OFF"
  Send "SET LED 2 OFF"
  Wait 1
EndFor
EndPrgm
```

**Challenge 2b:** Try Challenge 2a once more using the "BLINK" and "TIME" options. This time you can do it without a For Loop.

- Example program:

```
Define pc2b()=
Prgm
Send "CONNECT LED 1 TO OUT 1 "
Send "CONNECT LED 2 TO OUT 2 "
Send "SET LED 1 ON BLINK 3 TIME 30"
```



```
Send "SET LED 2 ON BLINK 3 TIME 30"
Wait 30
EndPrgm
```

**Challenge 3:** Connect a continuous servo motor to the TI-Innovator Hub and cause it to rotate clockwise (CW) and then in the opposite direction, counterclockwise (CCW).

**Teacher Guidance during Challenge 3:**

- Use a For..EndFor loop to cause a continuous servo motor to rotate clockwise and then counterclockwise.
- Use Send "CONNECT SERVO.CONTINUOUS # TO OUT 3"
- The SERVO.CONTINUOUS motor must be connected to the OUT 3 port because 5 volts are necessary to power the servo motor.
- Use Clockwise (CW) or Counter Clockwise (CCW) to set the turn direction of the continuous servo.
- Power for the continuous servo motor is ranged from 0 to 100; 0 being off and 100 being full power.
- Be sure external battery is attached to the PWR port of the TI-Innovator Hub and turned on just before running program.
- Example program

```
Define c3( )=
Prgm
Send "CONNECT SERVO.CONTINUOUS 1 TO OUT 3"
For n,1,5
  Send "SET SERVO.CONTINUOUS 1 CW 20 TIME 1"
  Wait 2
  Send "SET SERVO.CONTINUOUS 1 CCW 20 TIME 1"
  Wait 2
EndFor
EndPrgm
```

- Experiment with changing the power and TIME duration and the associated wait time to see what effect that has on how many turns the servo will make.

**Challenge 4:** Connect a temperature sensor to the TI-Innovator Hub and display the temperature on the calculator.

**Teacher Guidance during challenge 4:**

- The Hub Connect commands provide the hub with information about the type of device plugged into external ports and how to communicate with that device.

- The Read command instructs the Hub to read a value from a device plugged into a specified port. The Hub places the value in a memory location on the Hub, waiting for the calculator to “get” the value.
- The Get command gets the value from the Hub memory location for values read and stores the value to a specified variable.
- While loops are useful when you would like a set of commands to be executed when a certain condition is true.
  - The While loop test checks to see if the variable named *key* is not equal to “esc”. If *key* is not equal to “esc” then the commands in the loop are run. If *key* is equal to “esc” then the program exits the While loop and moves to the next command after the EndWhile statement. In the example, the variable named *key* is initially set to be a blank string. This assures that the While loop will run at least once.
  - In the While loop a function called getKey() is used to update the value stored in the variable named *key*. getKey() is a TI-Nspire function that is useful for keyboard controlled While loops. getKey() stores the name of the last key that you pressed while a program is running. The key names are text string values, which are enclosed in quotes. For example, the While loop test checks for “esc” not esc.
- Note that multiple arguments can be used to build a message with the DispAt command.
- Example program:

```
Define c4 ()=  
Prgm  
Send "CONNECT TEMPERATURE 1 TO IN 1"  
DispAt 8,"Press [esc] to Quit"  
key:=""  
  
While key≠"esc"  
  Send "READ TEMPERATURE 1"  
  Get t  
  DispAt 3,"Temperature=",t,"°C"  
  Wait 0.5  
  key:=getKey()  
EndWhile  
EndPrgm
```



**Challenge 5:** Connect the Hall effect magnetic proximity sensor, which determines if the south pole of a magnetic field is close to the sensor. Display “Magnet is present” or “Magnet is not present” based on the reading of the Hall effect sensor and the position of the magnet.

**Teacher Guidance during Challenge 5:**

- Some sensors or output devices do not have an associated object that can be found in the input and output menus. For these devices we use ANALOG.IN, DIGITAL.IN, ANALOG.OUT, or DIGITAL.OUT.
- The Hall effect sensor detects the presence (or absence) of a magnetic field with south polarity. For the Hall sensor you will use ANALOG.IN.
- Sensor values are related to voltage measurements. When the south pole of the magnet is close, readings are usually under 100.
- It is recommended that IN3 is used to supply enough power to the sensor.
- Note: The © symbol denotes a comment. Comments are not executed when the program is run. Use the Program/Actions menu to add a comment.
- Example program:

```
Define c5()=
Prgm
Send "CONNECT ANALOG.IN 1 TO IN 3"
© ANALOG.IN 1 is the object for the Hall effect magnetic proximity sensor
DispAt 8,"Press [esc] to Quit"
key:=""
While key≠"esc"
    Send "READ ANALOG.IN 1"
    Get m
    DispAt 3,"Magnetic sensor(bits)= ",m
    If m<100 Then
        DispAt 5,"Magnet is present"
    Else
        DispAt 5,"Magnet is not present"
    EndIf
    Wait 0.5
    key:=getKey()
EndWhile
EndPrgm
```





**Final Challenge:** Use the skills developed in the earlier challenges to develop an alarm system for a model of a car that determines if a pet is present (magnet) AND the temperature reading inside the car reaches a critical threshold before triggering the alarm.

**Teacher Guidance during Final Challenge**

:

- This challenge has students create a system that reads data from inputs such as temperature and the hall sensor which in turn controls LED's, sounds through the speaker, and the continuous servo motor.
- Notice that While..EndWhile is used often with sensors (inputs) and If..EndIf is used with outputs.
- Example program:

```
Define c6()=
Prgm
Send "CONNECT TEMPERATURE 1 TO IN1"
Send "CONNECT TEMPERATURE 2 TO IN2"
Send "CONNECT ANALOG.IN 1 TO IN3"
Send "CONNECT LED 1 TO OUT1"
Send "CONNECT LED 2 TO OUT2"
Send "CONNECT SERVO.CONTINUOUS 1 TO OUT3"
DispAt 1,"Pet Car Alarm is ON"
DispAt 8,"Hit ESC to quit monitoring car"
key:=""
window:=1
While key≠ "esc"
  Send "READ TEMPERATURE 1"
  Get temp_in
  Send "READ TEMPERATURE 2"
  Get temp_out
  Send "READ ANALOG.IN 1"
  Get m
  If m<100 Then
    DispAt 3,"Pet in car"
  Else
    DispAt 3,"Pet not in car"
  EndIf
  If window=1 Then
    DispAt 4,"Window is closed"
```



```
Else
DispAt 4,"Window is open"
EndIf
DispAt 5,"Temperature inside car= ",temp_in,"°C"
DispAt 6,"Temperature outside car= ",temp_out,"°C"
If temp_in>25 and m<100 Then
  DispAt 2 "Warning! Pet is in heat distress!"
  Send "SET LED 1 ON BLINK 3"
  Send "SET LED 2 ON BLINK 3"
  Send "SET SOUND 440 TIME .5"
  Wait 0.5
  Send "SET SOUND 880 TIME .5"
  Wait 0.5
  If window=1 Then
    Send "SET SERVO.CONTINUOUS 1 CW 30 .25"
    Wait .25
    window:=0
  EndIf
Else
  DispAt 2 "No warning"
  Send "SET LED 1 OFF"
  Send "SET LED 2 OFF"
  Send "SET SOUND OFF"
  If window=0 Then
    Send "SET SERVO.CONTINUOUS 1 CCW 30 .25"
    Wait .25
    window:=1
  EndIf
EndIf
key:=getKey()
EndWhile
Send "SET LED 1 OFF"
```



## Challenge: Pet Car Alarm Project

TI-NSPIRE™ CX

TI-INNOVATOR™ STEM PROJECT

TEACHER NOTES

```
Send "SET LED 2 OFF"  
Send "SET SOUND OFF"  
Send "SET SERVO.CONTINUOUS 1 0"  
DispAt 1,"Pet Car Alarm is OFF"  
DispAt 2," "  
EndPrgm
```